REMARKS

Claim 18-37 are pending in this application. The Examiner has rejected claims 18-37 under 35 U.S.C. §102 as being anticipated by U.S. Patent No. 6,223,540 ("*Schaumont*") issued to Schaumont et al. In the Advisory Action dated March 5, 2003, the Examiner states that *Schaumont* discloses the limitation "assigning a type and dimension to an undeclared variable by analyzing the functional description to form an abstract syntax tree." Although Applicants disagree that *Schaumont* discloses assigning type and dimension to undeclared variables by analysis of the design, Applicants have amended the claims to further clarify the differences between the present invention and *Schaumont*.

The Examiner cites FIG. 11 and column 24 of *Schaumont* as assigning types and dimensions to undeclared variables. FIG. 11 discloses the design of a finite state machine description. In particular, the discussion of FIG. 11 in the specification describes declaring a type "T_sample" to use a specific bit width. *Schaumont* 18:21-34. The initial type declaration either use a default bit width (in which case all types will have the same arbitrarily-chosen bit width) or can be explicitly declared by the designer. *Schaumont* 18: 21-29. Then, the designer may use these declared types in defining variables, such as "coef":

sig_array coef (4, ck, T_sample)

Schaumont, FIG. 11. The system disclosed in *Schaumont* will then assign the bit width declared in "T_sample" to the elements of the "coef" array. Notably, neither the declaration of the type, nor the assignment of the type to the array variable occurs through analysis of the usage of the variable in the design. Rather, the designer performs the declaration and assignment.

In contrast to merely assigning a pre-declared bit width to a variable, the claimed invention requires that the assignment of types and dimensions for undeclared variables be performed automatically by the system. The invention ***analyzes*** how the undeclared variables are used in the design and ***infers*** the appropriate type and dimension from their usage. Thus, the designer need not concern herself with determining the proper size for each variable appearing in a MATLAB design because the system itself performs this analysis and assignment.

To further clarify this aspect of the invention, applicants have amended the claims as follows. Amended claims 18 and 33 now recite the following limitations:

> ***inferring*** a type and dimension for the undeclared variable by ***analyzing the usage*** of the undeclared variable in the abstract syntax tree;
>
> assigning the ***inferred*** type and dimension to the undeclared variable;

Additionally, amended claim 26 recites "a type-shape analyzer, coupled to the parser, for ***inferring*** a type and a dimension to the undeclared variable by ***analyzing*** use of the undeclared variable in the abstract syntax tree." These three claims require that the system itself analyze the usage of undeclared variables, infer the proper type and dimension, and then assign the proper type and dimension to the undeclared variables. These limitations require more than *Schaumont's* simple substitution of a pre-declared type for a label in a variable declaration as discussed above. Thus, the claims as amended are patentably distinct from *Schaumont.*

Furthermore, the Examiner has cited FIG. 22, 23 and column 24 as disclosing using "C++ objects such as signals and finite state machine descriptions which translate themselves to a control/data flow data structure." The Examiner does not indicate how this applies to the present invention or how this statement anticipates the claims. As Applicants discussed in Amendment B, FIG. 22, 23 and column 24 do not teach or disclose ***inferring*** a type and dimension for the

undeclared variable by ***analyzing*** the ***usage*** of the undeclared variable as required by the independent claims. Column 24 describes the use of compiled and interpreted C++ descriptions in achieving the dual goals of simulation and synthesis. This high-level discussion is irrespective of whether the C++ description includes declared or undeclared variables or even requires type-dimension analysis. Although the cited portions of the specification discuss storing the C++ description in memory, it does not disclose or teach that the stored C++ description has undeclared variables or that analysis of the usage of variable is performed for the purposes of inferring the dimension and type of the undeclared variables.

Moreover, the code generation step described in *Schaumont*, Column 24, simply cannot contain undeclared variables in a C++ description because at this point in the design flow, the type and dimension of every variable in the description has already been assigned. For example, "HDL code generation requires the C++ description to be available as a data structure that can be processed by the code generator." *Schaumont* 24:20-22. For the C++ description to be available as a data structure, the sizes of each variable must already be declared, otherwise it could not be stored as a data structure, nor could it be synthesized into HDL code. Additionally, *Schaumont* describes the C++ description as comprised of objects such as the finite state machine discussed above in column 18: "The description uses C++ objects such as signals and finite state machine descriptions which translate themselves to a control/data flow data structure." *Schaumont* 24:26-29. As discussed above, these objects already contain declared variables with assigned types, and as such, no type-dimension analysis is performed. Thus, the code generation and synthesis steps in *Schaumont* do not disclose or teach inferring type and dimensions of undeclared variables by analyzing usage as required by the independent claims.

Claims 19-25 depend from 18, claims 27-32 depend from 26, and claims 34-37 depend from claim 33. As discussed above, claims 18, 26 and 33 are patentable, and as such, claims 19-25, 27-32, and 34-37 are also patentable for at least the same reasons. Moreover, these dependent claims are patentable for additional reasons. For example, claims 20 and 35 recite "analyzing the value range of the at least one undeclared variable; and assigning the required precision," and claim 28 recites "a precision analyzer." As discussed above, *Schaumont* does not disclose or teach a system that analyzes undeclared variables to infer attributes such as type, dimension, and precision. As such, *Schaumont* does not anticipate the dependent claims.

Conclusion

In sum, Applicants respectfully submit that claims 18-37 as presented herein, are patentably distinguishable over the cited references (including references cited, but not applied). Therefore, Applicants request consideration and allowance of these claims.

Applicants respectfully invite Examiner to contact Applicants' representative at the number provided below if Examiner believes it will help expedite furtherance of this application.

RESPECTFULLY SUBMITTED,
Prithviraj Banerjee, Alok Choudhary, Malay Haldar, Anshuman Nayak

Date: 06/03/04 By: Deepti Panchawagh

Deepti Panchawagh-Jain, Esq.
Registration No. 43,846
The Law Office of Deepti Panchawagh-Jain
3039 Calle De Las Estrella
San Jose, CA 95148
(408) 506-5352 (Phone)
(801) 838-1085 (Fax)